



Assemblage d'objets 3D Semi-Automatique basé géométrie en Archéologie

Nicolas Mellado, Patrick Reuter, Christophe Schlick

► To cite this version:

Nicolas Mellado, Patrick Reuter, Christophe Schlick. Assemblage d'objets 3D Semi-Automatique basé géométrie en Archéologie. 22èmes Journées de l'Association Française d'Informatique Graphique (AFIG), Nov 2009, Arles, France. p141. hal-00576904

HAL Id: hal-00576904

<https://hal.science/hal-00576904>

Submitted on 15 Mar 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Assemblage d'objets 3D Semi-Automatique bas gomtrie en Archologie

Nicolas Mellado¹ et Patrick Reuter² et Christophe Schlick²

¹INRIA Bordeaux (Projet Iparla), ²INRIA Bordeaux (Projet Iparla) - Universit Bordeaux 2

Abstract

The recent use of 3D laser scanning in the archaeological context requires techniques and systems to make the generated data usable. In this paper, we focus on the fragment reassembly based on ArcheoTUI, a tangible interface created to assemble the digitals models of fragments manually. We present a semi-automatic system to assist the user in real time in when assembling the fragments. This approach allows to use both the accuracy of geometric algorithms and the scientific knowledge of the archaeologist. The user specifies an approximate initial position of the two fragments by means of the tangible interface. Then, our system performs a real time registration thanks to the use of the Iterative Closest Point registration algorithm combined with a speed-up data structure and a bi-factorial weighting of the pairs. The data structure is used to find an area of interest on each object and to accelerate the pairs generation, whereas the weighting improves the detection of the data relevance.

L'utilisation rcnte de l'acquisition par scanner laser en archologie ncessite des techniques et des systmes permettant de rendre les donnes cres exploitables. La problmatique aborde dans cet article est le r-assemblage de fragments d'objets faisant suite au systme ArcheoTUI, qui consiste en une interface tangible pour manipuler les modles numriques de fragments et les assembler manuellement. Dans ce papier, nous prsentons un systme semi-automatique permettant d'assister en temps rel l'utilisateur pour effectuer le r-assemblage des fragments. Cette approche permet d'associer la prcision du calcul gomtrique aux connaissances scientifiques de l'archologue. L'utilisateur positionne au plus proche les objets deux deux via l'interface tangible. Ensuite, notre systme ralise l'assemblage des fragments en temps rel en utilisant l'algorithme d'assemblage Iterative Closest Point associ une structure de donnes acclratrice et une pondration des paires bi-facteurs. La structure de donnes est utilise pour dtecter des zones d'intrt sur les objets et pour acclrer la cration des paires tandis que la pondration permet une meilleure dtection de la pertinence des donnes.

1. Introduction

La rsolution de puzzles 3D est une problmatique rcurrente en informatique graphique. La majorit des cas d'applications sont lis l'archologie, domaine permettant de manipuler des donnes non synthtiques ncessitant des algorithmes plus robustes que les cas d'cole. L'orientation des techniques actuelles est axe sur le tout-automatique. Cependant, si l'on prend en compte que ces algorithmes peuvent tre utilis sur des donnes complexes et relles, on augmente de manire importante la difficult de la tche : les informations sman-

tiques transmettre entre l'utilisateur et le systme peuvent tre trs complexes et difficile paramtrer. De plus, l'absence d'implication de l'utilisateur dans le processus de rsolution oblige les algorithmes retrouver toutes les conclusions sur le puzzle que l'utilisateur peut connaitre grce ses connaissances implicites. Enfin, la dgradation des fragments par l'rosion et la possibilit de trous dans le puzzle complexifie davantage le problme.

Dans ce papier nous prsentons une approche de rsolution de puzzle semi-automatique permettant une

association des compétences de l'utilisateur et des algorithmes. Nos travaux sont principalement axés sur l'optimisation des algorithmes d'assemblage pour permettre un fonctionnement en temps réel, associant une technique de pondération des données permettant d'augmenter la robustesse des calculs. Ces opérations sont réalisées dans un contexte d'assemblage d'objets deux-deux, en tenant en compte uniquement des informations géométriques locales à la zone d'assemblage.

2. État de l'art

2.1. Assemblage géométrique

L'assemblage géométrique est beaucoup utilisé en informatique graphique et en vision robotique, notamment pour le recalage de surfaces acquises par scanner laser, l'assemblage d'objets virtuels, la reconnaissance et la classification de modèles. Si ce dernier domaine d'application nécessite des approches globales souvent liées des paramétrisations des modèles [WCW06, BDTK07], la résolution de puzzles archéologiques nécessite au moins une approche locale. En effet, les objets sont initialement représentés sous forme de nuages de points très denses nécessitant une sélection des données pertinentes pour éviter de surcharger le système de résolution avec des données inutilisables et potentiellement introduire une erreur.

Présenté en 1992 par Besl dans [BM92], l'ICP (pour *Iterative Closest Point*) est l'algorithme de référence pour le recalage de surfaces. Il s'agit d'un algorithme itératif qui fait converger au fur et à mesure des itérations deux jeux de données rigides l'un vers l'autre. Il utilise pour cela en entrée les sous-ensembles de surface du modèle scannés depuis des points de vue différents. Le positionnement relatif de ces jeux de données permet de calculer une erreur qu'il faut minimiser pour rapprocher les deux ensembles. Rusinkiewicz et al. proposent une taxonomie des différentes variantes de l'ICP [RL01] ainsi qu'une nouvelle signification au sigle ICP : *Iterative Corresponding Point*.

2.2. Recalage temps réel

L'introduction d'une contrainte temps réel pour l'ICP nécessite de faire des choix sur les techniques utilisées pour faire un compromis entre le volume des données et le temps nécessaire à leur traitement. Rusinkiewicz et al. proposent dans [RHHL02] une chaîne complète de recalage en temps réel basée sur l'ICP.

Le premier objectif est de ne prendre en compte que les points importants et utiles au recalage sans pour autant que le pré-traitement de sélection soit plus lourd que l'exploitation des points dans la chaîne. Dans [SDG07], les auteurs proposent une variante de l'ICP,

basée sur une hiérarchie de volumes englobants, nommée Trimmed ICP, permettant de recalibrer des objets sans prendre en compte de position initiale. Outre le fait de supprimer la notion de position initiale, cette variante introduit une méthode de recherche de zone d'entrée, qui permet de ne pas considérer les points inutiles du module de *r*-échantillonnage (donc avant le goulet d'entrave), au lieu de les traiter jusqu'au module de suppression des paires.

Le second objectif est lié au goulet d'entrave de l'algorithme de l'ICP : la création des paires. Outre le fait qu'elle doit être opérée pour tous les points sélectionnés de l'objet principal (d'où le point précédent), elle nécessite l'utilisation de structures de données pour accélérer la recherche spatiale [Sim96] ou pour simplifier l'accès aux données [BL95, BS97].

Enfin, de nombreuses méthodes existent pour augmenter la vitesse de convergence de l'ICP par la pondération des paires [GRB94], leur filtrage [RL01] et enfin le calcul de l'erreur [CM91] pour diminuer son temps d'exécution.

2.3. Taxonomie

Nous proposons le classement suivant pour les méthodes de résolution : méthodes manuelles, semi-automatiques et automatiques (voir figure 1).

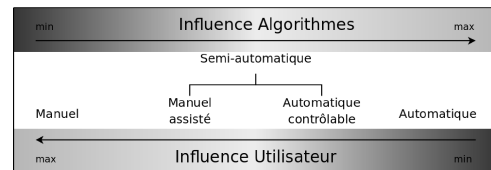


Figure 1: Classement des systèmes de résolution.

Les approches manuelles ne prennent en compte que les informations entrées par l'utilisateur : nature des fragments, transformations géométriques et données topologiques entre autres. La résolution du puzzle dépend exclusivement des capacités et des connaissances de la personne.

Les approches semi-automatiques intègrent dans les méthodes manuelles un ensemble de techniques dont le but est de résoudre, partiellement ou complètement, le même problème que l'utilisateur. Nous distinguons les méthodes manuelles guidées où le système aide l'utilisateur et les méthodes automatiques contrôlables dans lesquelles l'utilisateur guide le système pour qu'il résolve le problème.

Les approches automatiques minimisent l'influence de l'utilisateur : il spécifie un paramétrage en une fois et le système propose une sortie. Des ajustements de

paramtrage peuvent tre faits, mais uniquement en fonction de la configuration finale calcule. Pour appliquer ces modifications, il est ncessaire de re-parcourir la chaine complte et d'observer le rsultat final. Gnralement les algorithmes ont besoin d'un grand nombre de paramtres pour pouvoir s'adapter plusieurs situations. Cela entrane souvent une difficult de paramtrage : les ajustements proposs l'utilisateur sont tellement nombreux qu'il peut tre difficile de trouver la bonne combinaison.

2.4. Systmes de rsolution de puzzles 3d archologiques

La plupart des solutions proposes dans la littrature pour cette problmatique sont bases sur des approches automatiques. Certaines mthodes sont plutt orientes pour des lments de faible taille, comme celle de Willis [WOC03] spcialise pour les lments connus comme des surfaces de rvolution. Brown et al. proposent dans [BTFN*08] une mthode pour la reconstruction de fresques, permettant une rduction du probleme par des approches 2.5D. Huang et al. [HFG*06] proposent quant eux une solution de r-assemblage complet d'objets statutaires, base sur une pr-segmentation pour dtecter des faces, une association des fragments deux deux, puis une rsolution globale pour toutes les relations d'adjacence.

Cette technique est amliore dans [TFK*09] pour permettre la reconstruction partielle d'un temple, en ajoutant des notions de contraintes sur les associations de fragments en fonction de motifs et de moulures. L'approche automatique assiste prsente dans [KTN*05] pour reconstruire un plan de Rome grave sur du marbre est aussi base sur la prolongation de lignes d'un fragment l'autre.

Toutes ces mthodes sont fonctionnelles sur des jeux de donnees restreints car elles n'exploitent pas ou peu les connaissances des archologues, contrairement aux approches semi-automatiques que nous prsentons ensuite.

3. Notre systme semi-automatique

3.1. Interface Manuelle : ArcheoTUI

Notre systme est construit autour de l'interface manuelle d'assemblage d'objets archologiques par des interfaces tangibles ArcheoTUI [RRC*07], qui permet l'utilisateur d'assembler virtuellement des fragments deux deux en vue de recomposer un objet global.

Le systme est bas sur des interfaces tangibles six degrs de libert (lments (1) et (2) sur la figure 2) auxquelles sont virtuellement associs les fragments

scans (lments (5) et (6) sur la figure 2). Les mouvements que l'utilisateur applique ces interfaces sont appliqus aux fragments pour permettre une correspondance directe entre les gestes raliss par l'utilisateur et le positionnement des objets virtuels.

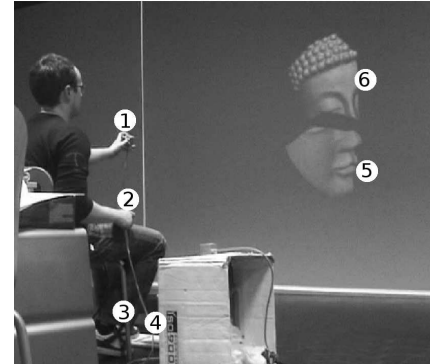


Figure 2: Interface du programme ArcheoTUI.

Afin de proposer une solution semi-automatique en temps rel, nous avons intrgr dans ce systme d'assemblage manuel des algorithmes d'assemblage et les structures associes selon le schma global suivant :

- **Pr-calculs** : lorsque le modle est charg dans la scne, toutes les structures de donnees statiques ncessaires au r-assemblage sont gnes,
- **Recalage** : chaque dplacement des objets, l'assistance est sollicite pour recalculer une configuration d'assemblage suppose localement optimale. C'est durant son excution que sont calcules les transformations supplmentaires appliquer aux objets pour les mettre en correspondance. Si la correspondance n'est pas possible, aucune transformation n'est calcule. Cela peut se produire si les objets sont trop loigns, ou non compatibles localement.

3.2. Algorithme de recalage

Notre systme d'assemblage a pour base l'ICP, algorithme itratif permettant de calculer les transformations rigide qui minimisent l'erreur entre deux ensembles de donnees. Chaque itration consiste slectionner les donnees utiliser, apparier les lments des deux ensembles, pondrer ces paires, calculer la distance entre les deux sous-ensembles slectionnns et enfin calculer les transformations rigides.

Cette conception modulaire fait de l'ICP un algorithme adaptable, que nous avons modifi comme suit :

- **Dtection de zone d'intrt** : cette phase permet de slectionner un sous-ensemble pour chaque nuage de points representant la zone utile au recalage en

considérant des intersections entre des hiérarchies de sphères englobantes.

- **Sélection des paires** : il s'agit d'associer chaque point d'un ensemble son homologue le plus proche dans l'autre ensemble,
- **Pondération et rejet des paires** : nous réalisons une pondération en fonction de la distance et de la compatibilité des normales entre les deux points de chaque paire. Positionner une pondération 0 revient rejeter la paire,
- **Minimisation de l'erreur** : cette phase revient calculer une transformation rigide qui minimise la distance entre les points des paires sélectionnées telle que définie dans [BM92].

3.3. Interaction avec l'utilisateur

Il est nécessaire que les actions de l'utilisateur aient une influence sur les données d'entrée des algorithmes, donc potentiellement leur sortie. Dans notre contexte d'application, l'influence de la position initiale sur le module de recalage est très importante. Comme nous sommes dans un contexte temps réel, il est nécessaire que les calculs soient effectués après chaque mouvement des pièces. Ainsi, lorsque l'utilisateur utilise l'une des interfaces tangibles, le système applique la transformation, calcule une position d'assemblage suppose localement optimale et affiche les informations de sortie.

L'utilisateur est considéré comme le "décideur" du système. Ainsi, toutes les actions effectuées par l'assistance sont visualisables et sont appliquées uniquement si l'utilisateur le décide. Plusieurs informations peuvent être utiles à l'utilisateur pour essayer d'affiner un positionnement après exécution de l'algorithme d'assemblage. Voici celles que nous avons retenues pendant notre développement.

Position calculée : nous affichons l'objet dans la position suppose localement optimale calculée par l'algorithme en transparence dans la scène. Cela permet de visualiser en plus des deux positions des objets, un objet recalé, et ainsi de conserver la métaphore interface tangible et fragment virtuel, et enfin de visualiser graphiquement la sortie de l'algorithme (voir figures 3 et 11 en annexe).

Erreur de sortie : l'erreur mesurée (somme des distances intra-paires au carré) est le critère d'arrêt et d'application de l'algorithme d'assemblage. L'affichage de la dernière valeur calculée indique à l'utilisateur la pertinence de la position proposée par le système.

4. Approche temps réel

Cette section présente les algorithmes et les structures de données utilisés pour d'une part améliorer la qualité

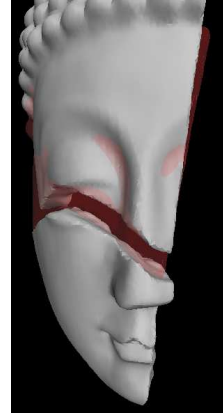


Figure 3: Affichage en transparence de la position d'assemblage suppose localement optimale calculée.

des données d'entrée, et d'autre part diminuer la quantité de données retenues, afin d'accroître les calculs. On entend par qualité le fait qu'une donnée soit exploitable et n'introduise pas d'erreur dans les calculs de minimisation.

Dans sa version originale, l'ICP utilise la totalité des points des deux objets pour effectuer le recalage : les deux nuages de points représentent la même surface. Dans le cas de l'assemblage de fragments, il est important de ne considérer que les points des deux objets qui sont dans la zone de contact : les paires contenant des points hors de cette zone sont forcément invalides et induisent une erreur.

L'autre problématique à prendre en compte concerne le module de création des paires par recherche du point le plus proche. Il est bien souvent le goulet d'étranglement de l'algorithme, quelles que soient les variantes ou les structures de données utilisées. C'est pour réduire le temps passé dans ce module que nous avons mis au point un module de sélection de zone d'entrée, qui permet de diminuer le nombre de points traités en fonction des positions initiales des fragments.

Il est important de noter que l'utilisation de structures de données et d'algorithmes en vue d'accroître la vitesse d'exécution doit être totalement transparente sur la qualité des données arrivant à l'étape finale de la minimisation. Il convient donc d'utiliser des algorithmes et des structures accélératrices plus permissifs que les algorithmes de sélection des données (pondération et rejet en fonction des normales et des distances).

4.1. Recherche de zone d'intrt

L'objectif du module de dtection de zone d'intrt est de trouver la zone de chaque objet exploitable pour calculer le recalage. Pour ce faire, nous utilisons une hiarchie de sphres englobantes pour raliser des tests de proximit entre les deux nuages de points. La recherche est base sur l'hypothse suivante : les points contenus dans deux sphres qui s'intersectent sont au plus distants de la somme des diamtres des sphres englobantes.

Cette approche ne permet pas elle seule de garantir de slectionner tous les points dont la distance d est infrieure un seuil d_T . Dans le cas o des points sont positionns proches de la limite de leur sphre englobante, ils peuvent tre proches des points de l'autre objet sans que les sphres s'intersectent (voir figure 4 o $D1 < d_T$). Pour corriger cela, nous utilisons l'equation d'intersection des sphres (1) qui permet d'assurer cette distance minimale (on note R_u et R_v les rayons des sphres englobantes, et d_{uv} la distance entre les deux centres).

$$d_{uv} \leq R_u + R_v + d_T \quad (1)$$

En pratique, cela revient gonfler le rayon des sphres d'une distance $\frac{d_T}{2}$, comme visible la figure 4. Cette technique est associe un rejet des paires dont la distance entre les points est suprieure au seuil.

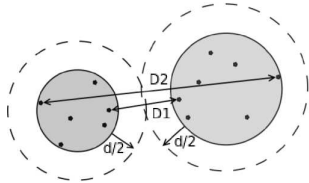


Figure 4: Configuration illustrant l'importance d'augmentation de la taille des sphres pour garantir la slection de tous les points en fonction d'une distance seuil.

Rechercher les zones d'intrt des modles revient trouver les sphres qui intersectent les sphres de l'autre objet. On effectue pour cela une descente en escalier dans les graphes : on rduit d'un niveau successivement dans un objet puis dans l'autre, et les sphres englobantes qui n'intersectent pas avec l'autre modle sont mises de cot. L'arrt s'effectue lorsque toutes les sphres slectionnes sont des feuilles du graphe, ou lorsque l'on atteint un seuil limite k , que l'on prcisera par la suite.

4.2. Optimisation de la cration des paires de points

Pour acclrer la recherche du point le plus proche, qui est ncessaire pour l'ICP, la hierarchie de sphre englobante est augmente par un kD-tree. En utilisant cette structure mixte, on rduit la complexit maximale de l'algorithme de $O(n^2)$ $O(n_s \log(n))$ o n_s est le nombre de points slectionns lors de la recherche de zone d'intrt du premier objet, et n le nombre de points du second. Il est possible de rduire encore davantage la complexit globale en modifiant les interactions entre les deux modules, comme suit.

L'algorithme de zone d'intrt ralise des tests d'intersection entre les sphres englobantes. Les sphres dans lesquelles se trouvent les points les plus proches du second objet sont connues ds cette tape : ce sont celles avec lesquelles il y a intersection (figure 5). Si

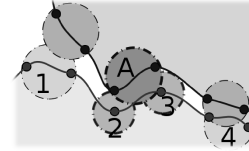


Figure 5: Rduction du sous-ensemble contenant le point le plus proche : les voisins des points prsents dans la sphre A sont forcement soit dans la sphre 2 soit dans la sphre 3.

l'on sauvegarde pour une sphre donne toutes celles de l'autre objet avec laquelle elle intersecte, on peut s'en servir comme point d'entre dans le kD-tree pour la recherche du point le plus proche.

Pour trouver les sphres qui s'intersectent, il est ncessaire de tester toutes les sphres entre elles. L'algorithme de recherche passe donc une complexit de $O(m^2)$, o m est le nombre de sphres slectionnes. L'algorithme de recherche du point le plus proche passe ensuite en temps constant : on ne slectionne que des feuilles lors de la recherche de zone d'intrt.

Dans notre domaine d'application, le pourcentage de recouvrement ρ entre les objets deux deux est suppos relativement faible : il y a contact par une ou plusieurs faces, mais pas la totalit. Le nombre de sphres slectionnes m dpend de ce pourcentage, selon la relation suivante :

$$m = \rho * \frac{n}{\Theta} \quad (2)$$

o Θ est le nombre de points prsents dans une feuille du graphe. On peut donc obtenir un ordre de grandeur de m en fonction de ρ :

$$\begin{aligned} \rho \rightarrow 0\% : & \quad m \ll n \Rightarrow m^2 \leq n \log(n) \\ \rho \rightarrow 100\% : & \quad m \rightarrow n \Rightarrow m^2 \gg n \log(n) \end{aligned} \quad (3)$$

On voit donc qu'en fonction du recouvrement, il peut être préférable d'utiliser soit les sphères comme point d'entrée au kD-tree ($\rho \rightarrow 0\%$), soit le nœud père du graphe ($\rho \rightarrow 100\%$).

Il est possible de paramétrer la transition entre les deux variantes en jouant sur une profondeur limite k dans l'algorithme de recherche de zone d'entrée (où K_{max} est la profondeur maximum du graphe pour un modèle) :

- $k = 0$: on prend directement la racine du graphe : aucun rejet de sphère. On recherche donc le point le plus proche pour tous les points ($\rho = 100\%$),
- $k = K_{max}$: rejet maximal des sphères dépendant du modèle ($\rho = \rho_m$),

5. Pondration

5.1. Pondration par distance

La sélection des points par la hiérarchie de sphères englobantes garantit de sélectionner tous les points dont la distance de la paire potentielle est inférieure à un seuil d_T major des rayons R_u et R_v des sphères englobantes de chaque point (équation 1). Effectuer une pondration par seuillage pour rejeter les paires dont la distance d est supérieure à d_T induit une discontinuité sur la sélection des paires par rapport aux mouvements des objets. En effet, une paire dont la distance est proche du seuil peut, pour de faibles variations des transformations géométriques appliquées aux objets, être tantôt sélectionnée tantôt rejetée. Ainsi, malgré une faible différence de position initiale, cette paire introduit une erreur importante pouvant entraîner une importante différence sur les transformations finales.

Pour s'assurer d'utiliser une fonction continue sur l'intervalle $[0; 1]$ tout en diminuant au maximum le temps de calcul, la pondration de la distance intra-paire d est réalisée par une projection des distances normalisées dans l'approximation gaussienne polynomiale présente dans [Wen95] de continuité C^1 :

$$\begin{cases} f_d(d) = \left(\left(\frac{d}{d_T} \right)^2 - 1 \right)^2 & \text{si } 0 \leq d \leq d_T \\ f_d(d) = 0 & \text{sinon} \end{cases} \quad (4)$$

5.2. Pondration par compatibilité des normales

Dans certaines configurations, il arrive que des points présents dans la zone d'entrée définie en fonction de la distance ne fassent pas réellement partie de la surface de contact finale. La figure 6 illustre l'une de ces configurations, dans laquelle les paires n'appartenant pas à la tranche de l'objet ont une influence très grande dans le calcul. En effet, leur distance intra-paire est très faible,

on peut donc penser que leur pondration sera nettement plus favorable que la majorité des paires de la tranche. Leur influence aura tendance à rapprocher les deux centres de gravité des deux ensembles de points, et ainsi bloquer le processus de minimisation.

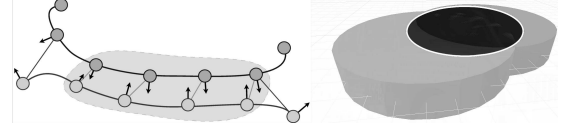


Figure 6: Deux cas pour lesquels tester la compatibilité des normales permet de détecter les paires qui ont été sélectionnées grâce à une faible distance entre leur point mais qui ne font pas réellement partie de la zone d'assemblage.

L'utilisation de la pondration en fonction de la cohérence des normales permet dans ce cas de figure de rejeter toutes les paires n'appartenant pas à la tranche : leurs normales ont tendance à être colinéaires de même sens, alors que les normales des paires de la tranche ont tendance à être opposées.

Tout comme pour la pondration en fonction de la distance, nous utilisons une approximation gaussienne pour réduire les discontinuités de sélection des paires. L'entrée utilisée pour le seuillage devant être un scalaire, nous utilisons comme variable d'entrée le produit scalaire entre les normales \vec{n}_i et \vec{n}_j .

On souhaite conserver uniquement les paires dont le produit scalaire est compris dans l'intervalle $[-1; -n_T]$. En combinant l'approximation gaussienne et une fonction de projection de l'intervalle $[-1; -n_T]$ à l'intervalle $[0; 1]$, on obtient la fonction f_n :

$$\begin{cases} f_n(\vec{n}_i, \vec{n}_j) = \left(\left(\frac{1 + \vec{n}_i \cdot \vec{n}_j}{1 - n_T} \right)^2 - 1 \right)^2 & \text{pour } -1 \leq \vec{n}_i \cdot \vec{n}_j \leq -n_T \\ f_n(\vec{n}_i, \vec{n}_j) = 0 & \text{sinon} \end{cases} \quad (5)$$

5.3. Pondration bi-varie

Notre système dispose de deux critères de pondration qu'il convient d'assembler pour réaliser une pondration efficace des paires. Comme on considère qu'une paire est valide uniquement si elle passe les deux filtres, et pour permettre à l'utilisateur de privilégier un type de seuillage par rapport à un autre, on peut utiliser un coefficient d'influence α associé aux équations (4) et (5) telles que :

- Si $0 \leq d_i \leq d_T$ ET $-1 \leq \vec{n}_i \cdot \vec{n}_j \leq n_T$

$$B_f(d_i, \vec{n}_i, \vec{n}_j, \alpha) = \frac{e\left(-\frac{\alpha}{f_d(d_i)}\right) * e\left(-\frac{1-\alpha}{f_n(\vec{n}_i, \vec{n}_j)}\right)}{e^{(-1)}} \quad (6)$$

- Sinon

$$B_f(d_i, \vec{n}_i, \vec{n}_j, \alpha) = 0$$

C'est le facteur α qui est accessible à l'utilisateur : une valeur faible favorise la pondération par la distance et une valeur élevée la pondération en fonction des normales. Le fait d'avoir les deux facteurs d'influence permet d'assurer une symétrie entre les deux pondérations (comme visible la figure 7). Nous avons gardé la possibilité de débrayer ce système en laissant la possibilité à l'utilisateur de ne choisir qu'une seule pondération.

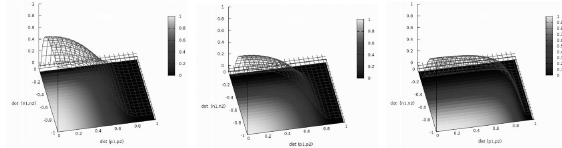


Figure 7: La variation du paramètre α permet de favoriser l'influence d'une propriété. De gauche à droite : on favorise la pondération par le produit scalaire ($\alpha = 0.9$), influence quiprovable ($\alpha = 0.5$), on favorise la pondération par la distance ($\alpha = 0.1$).

6. Résultats expérimentaux

6.1. Protocole

Cette section présente les résultats des tests que nous avons fait passer au système pour valuer ses capacités, avec comme processeur un Intel Pentium4 3.0Ghz et un système d'exploitation Linux. Pour cela, nous avons considéré trois scènes aux propriétés différentes :

Head Cette scène est constituée de deux fragments issus des modèles utilisés par Huang et al. dans [HFG*06], sous-échantillonnés environ d'un facteur 10 pour produire une scène légère.

Brick Cette scène est constituée de trois fragments eux aussi issus des modèles utilisés par Huang et al. dans [HFG*06], sous-échantillonnés environ d'un facteur 5 pour produire une scène plus lourde que Head. Deux fragments sont déjà assemblés et le test consiste à procéder à l'ajustement d'un troisième.

Fractal Cette scène est constituée de données totalement synthétiques générées à partir d'un algorithme fractal. Les deux objets sont strictement identiques et se recouvrent complètement.

Scène	Nb Sommets	Recouvrement	
		%	Nb Sommets
Head	≈ 24000	≈ 5%	≈ 1200
Brick	≈ 93000	≈ 10%	≈ 9300
Fractal	≈ 33000	100%	≈ 33000

Figure 8: Propriétés des trois scènes de test.

6.2. Analyse de la sélection des points et de la création des paires

Nous avons exprimé plusieurs valeurs de profondeur de graphe limite k pour la recherche de zone d'entrée afin de tester son influence sur les performances réelles de l'algorithme. La valeur maximale 1000 permet d'atteindre toutes les feuilles du graphe pour chaque modèle. Les résultats sont présentés à la figure 9. Le pourcentage de temps passé dans cette première phase est d'autant plus important que k est grand pour tous les modèles.

On peut cependant distinguer deux comportements en analysant les variations du temps d'exécution de l'algorithme. Pour les deux premiers modèles, Head et Brick, le temps d'exécution diminue d'autant plus que la recherche de zone d'entrée est réalisée en profondeur. Les faibles taux de recouvrement des objets entraînent un passage dans la première situation de l'équation (3) : $m \ll n$, avec une complexité critique de $n \log(n)$ pour la création des paires. Pour le troisième modèle Fractal, le taux de recouvrement maximal entraîne un comportement inverse. En effet, la complexité théorique est de l'ordre m^2 avec $m \rightarrow n$, on obtient donc une courbe strictement croissante en fonction de k .

L'utilisation de la variante consistant à utiliser les sphères comme point d'entrée dans le graphe pour la création des paires permet d'éviter la forte augmentation du temps de calcul lorsque l'on passe dans le cas 2 de l'équation (3) pour le troisième modèle.

6.3. Influence de la pondération

Les paires ont une double influence sur la convergence de l'ICP : elles sont utilisées d'une part pour calculer la minimisation, et d'autre part pour calculer l'erreur, critère d'arrêt de l'algorithme, indiquant la pertinence de cette minimisation. Si ces paires introduisent une erreur, la valeur d'erreur mesure et donc le test d'arrêt de l'algorithme ne sont plus fiables.

Ainsi, comparer la convergence pour plusieurs pondérations ne signifie pas grand chose : il est possible de converger dans plusieurs configurations, sans que le résultat ne soit valide pour l'archologue. L'analyse quantitative de l'influence de la pondération sur l'ICP nécessite la mise au point d'une mesure d'erreur ind-

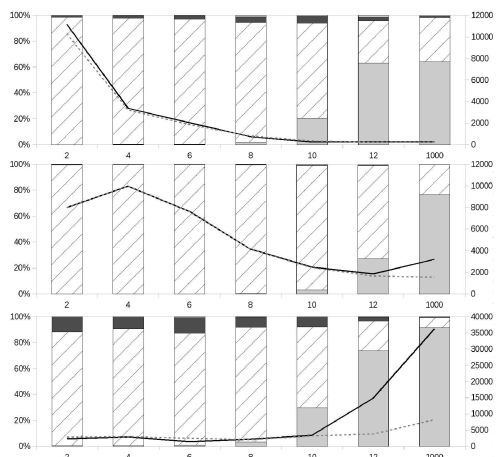


Figure 9: *évolution du temps de calcul (en millisecondes) en fonction de la profondeur maximale autorisée dans l'algorithme de recherche de zone d'intrt. De haut en bas, modèles Head, Brick et Fractal.*

pendante de l'ICP, prenant en compte la géométrie globale et locale des objets.

On peut cependant noter que pour les modèles Head et Brick, l'utilisation de la pondération par les normales a tendance à réduire le nombre de cas où l'ICP entraîne un positionnement contenant des intersections entre les modèles.

7. Discussion et travaux futurs

Le travail réalisé pour ce papier a permis de mettre en place un système semi-automatique d'assemblage de fragments orienté manuel assisté.

L'assemblage de fragments est une problématique souvent abordée par des approches automatiques : l'utilisateur transmet au système un ensemble de fragments et celui-ci se charge de réaliser la reconstruction. Ces approches ont le défaut de ne pas prendre en compte les connaissances *a priori* de l'utilisateur pour orienter la résolution. C'est dans cette optique que nous avons exploité une approche semi-automatique, permettant d'associer les compétences de l'utilisateur et les capacités de calcul du système.

Notre approche est basée sur l'algorithme de recalage ICP [BM92]. De nombreuses variantes de cet algorithme existent déjà : sa conception modulaire est propice à la création de variations dont nous nous sommes inspirés pour proposer deux nouvelles approches.

La première approche est la recherche de zone d'intrt, basée sur l'exploitation d'une hiérarchie de sphères englobantes combinée à un kd-tree. L'assemblage de fragments est un contexte très propice à cette recherche

car la majorité des données définissant un fragment ne sont pas utiles pour chaque association. Cette approche conservatrice peut néanmoins être utilisée pour l'application originelle de l'ICP : le recalage de surface. De plus, la recherche du plus proche voisin tant accélérée par le kd-tree, nous proposons deux variantes d'interaction entre la recherche de zone d'intrt et la recherche du point le plus proche pour optimiser l'une des deux tapes en fonction des caractéristiques des fragments.

La seconde approche consiste à considérer les normales aux points et la distance pour calculer une pondération bi-facteur des paires. Celle-ci est calculée à partir d'approximations gaussiennes [Wen95] pour chaque attribut combiné pour produire une fonction de pondération drivable. Cette propriété est très importante car elle permet de supprimer les discontinuités sur les variations des données d'entrée en fonction du temps, et par conséquent sur les sorties de l'algorithme. Le fait de combiner plusieurs informations pour calculer la pondération permet d'augmenter la robustesse de l'algorithme. Il est ainsi tout fait envisageable d'acquérir de nouveaux attributs sur la surface des objets et de s'en servir pour orienter les calculs.

Le système ainsi produit permet d'effectuer des assemblages de fragments en temps réel pour des scènes dont le nombre de points dans la surface de recouvrement ne dépasse pas 1200 points. Sa conception modulaire permet de tester de nouvelles techniques tout en bénéficiant de l'approche semi-automatique lors de travaux futurs.

La limitation actuelle du système est liée en grande partie au contexte d'application et au type d'algorithmes choisis : l'ICP permet de considérer uniquement des données locales. L'assemblage de fragments nécessite de prendre en compte la géométrie globale des objets pour être réellement efficace. Dans la même logique, l'erreur calculée en fonction des distances intra-paires pose problème : sa conception purement locale empêche la détection d'incompatibilités entre les objets à une échelle plus globale.

Dans les travaux futurs, nous nous focalisons sur la mise au point d'une métrique plus pertinente permettant de proposer des solutions vraisemblablement plus fiables que l'utilisateur peut juger de manière objective et qui améliore donc considérablement l'efficacité du système semi-automatique.

References

- [BDTK07] BLERIM M., DAVCEV D., TRJAKOVIC V., KALAJDZISKI L.: Spherical mapping : a powerful tool for 3d object matching. In *Proc. SETIT 2007* (2007).
- [BL95] BLAIS G., LEVINE M. D.: Registering multiview

- range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (1995), 820–824.
- [BM92] BESL P., MCKAY N.: A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (1992), 239–256.
- [BS97] BENJEMAA R., SCHMITT F.: Fast global registration of 3d sampled surfaces using a multi-z-buffer technique. In *Image and Vision Computing* (1997), pp. 113–120.
- [BTFN*08] BROWN B. J., TOLER-FRANKLIN C., NEHAB D., BURNS M., DOBKIN D., VLACHOPOULOS A., DOUMAS C., RUSINKIEWICZ S., WEYRICH T.: A system for high-volume acquisition and matching of fresco fragments: Reassembling Thera wall paintings. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 27, 3 (Aug. 2008).
- [CM91] CHEN Y., MEDIONI G.: Object modelling by registration of multiple range images. *Image Vision Comput.* 10, 3 (1991), 145–155.
- [GRB94] GODIN G., RIOUX M., BARIBEAU R.: Three-dimensional registration using range and intensity information. El-Hakim S. F., (Ed.), vol. 2350, SPIE, pp. 279–290.
- [HFG*06] HUANG Q.-X., FLÖRY S., GELFAND N., HOFER M., POTTMANN H.: Reassembling fractured objects by geometric matching. *ACM Trans. Graphics* 25, 3 (2006), 569–578.
- [KTN*05] KOLLER D., TRIMBLE J., NAJBBERG T., GELFAND N., LEVOY M.: Fragments of the city: Stanford's digital forma urbis romae project. In *Proceedings of the Third Williams Symposium on Classical Architecture* (2005).
- [RHHL02] RUSINKIEWICZ S., HALL-HOLT O., LEVOY M.: Real-time 3D model acquisition. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 21, 3 (July 2002), 438–446.
- [RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the icp algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling* (2001), pp. 145–152.
- [RRC*07] REUTER P., RIVIERE G., COUTURE N., SORRAING N., ESPINASSE L., VERGNIEUX R.: A tangible user interface for the virtual reassembly of fractured archaeological objects. In *Proceedings of VAST 2007* (2007), Eurographics.
- [SDG07] SYNAVE R., DESBARATS P., GUEORGUEVA S.: Automated trimmed iterative closest point algorithm. In *Advances in Visual Computing* (2007), vol. 4842 of LNCS, Springer.
- [Sim96] SIMON D.: *Fast and Accurate Shape-Based Registration*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1996.
- [TFK*09] THUSWALDNER B., FLÖRY S., KALASEK R., HOFER M., HUANG Q.-X., THÜR H.: Digital anastylis of the octagon in ephesos. *J. Comput. Cult. Herit.* 2, 1 (2009), 1–27.
- [WCW06] WANG D., CUI C., WU Z.: Matching 3d models with global geometric feature map. pp. 4 pp.+.
- [Wen95] WENDLAND H.: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.* 4 no. 4 (1995), 389–396.
- [WOC03] WILLIS A., ORRIOLS X., COOPER D. B.: Accurately estimating sherd 3d surface geometry with application to pot reconstruction. *Computer Vision and Pattern Recognition Workshop 1* (2003), 5.

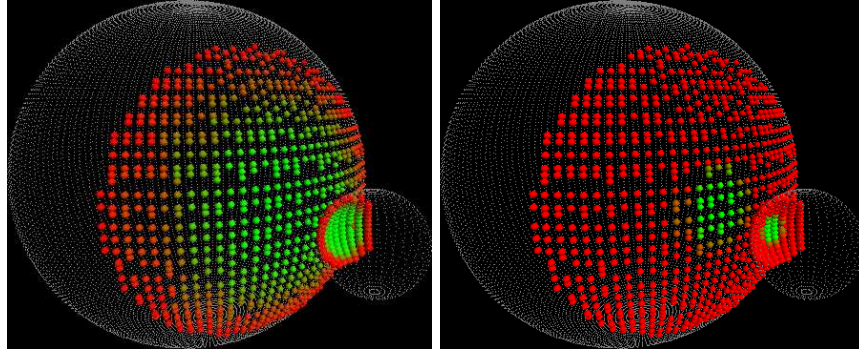


Figure 10: Influence du seuil n_T sur la pondration des paires d'objets sphériques. A gauche, tous les produits scalaires ngatifs sont autoriss ($n_T = 0$), droite seulement les produits scalaires compris dans l'intervalle $[-1; -0.8]$ ($n_T = 0.8$). En vert pondration 1, en rouge 0.

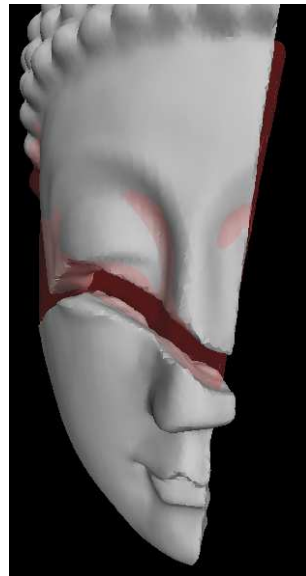


Figure 11: Affichage en transparence de la position d'assemblage calcule.

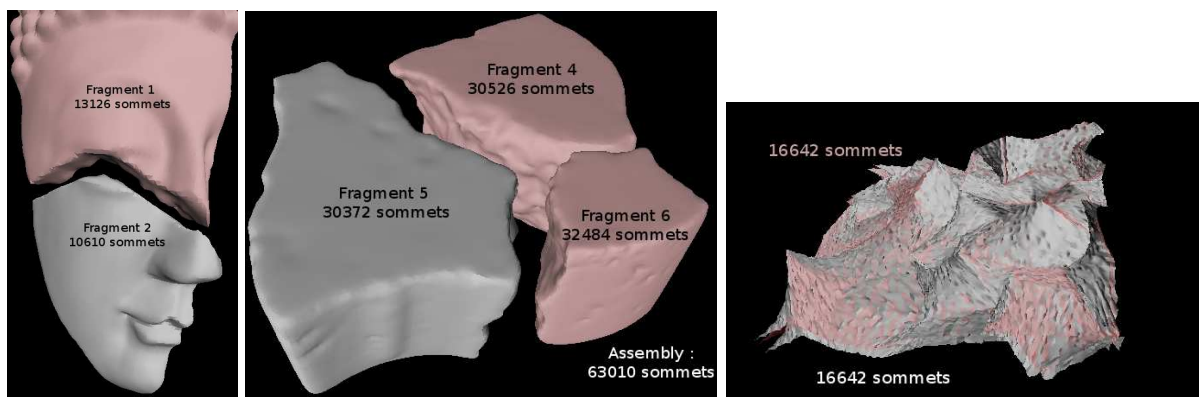


Figure 12: Vue des scnes de test Head, Brick et Fractal.